

00180160

Please type a plus sign (+) inside this box → ☒Approved for use through 09/30/2000. OMB 0651-0032  
Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE  
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.**UTILITY  
PATENT APPLICATION  
TRANSMITTAL**

Attorney Docket No. CS-20-000908

First Inventor or Application Identifier Shinya MATSUMOTO et al.

Title METHOD OF AND APPARATUS FOR RENDERING IMAGE,  
RECORDING MEDIUM, AND PROGRAM

Express Mail Label No.

(Only for new nonprovisional applications under 37 C.F.R. § 1.53(b))

**APPLICATION ELEMENTS**

See MPEP chapter 600 concerning utility patent application contents.

1. ☒ \* Fee Transmittal Form (e.g., PTO/SB/17)  
(Submit an original and a duplicate for fee processing)
2. ☒ Specification [Total Pages 30]  
(preferred arrangement set forth below)
- Descriptive title of the Invention
  - Cross References to Related Applications
  - Statement Regarding Fed sponsored R & D
  - Reference to Microfiche Appendix
  - Background of the Invention
  - Brief Summary of the Invention
  - Brief Description of the Drawings (if filed)
  - Detailed Description
  - Claim(s)
  - Abstract of the Disclosure
3. ☒ Drawing(s) (35 U.S.C. 113) [Total Sheets 18]

4. Oath or Declaration [Total Pages 5]
- a. ☒ Newly executed (original or copy)
- b. ☐ Copy from a prior application (37 C.F.R. § 1.63(d))  
(for continuation/divisional with Box 16 completed)
- i. ☐ **DELETION OF INVENTOR(S)**  
Signed statement attached deleting  
inventor(s) named in the prior application,  
see 37 C.F.R. §§ 1.63(d)(2) and 1.33(b).

**NOTE FOR ITEMS 1 & 13: IN ORDER TO BE ENTITLED TO PAY SMALL ENTITY  
FEES, A SMALL ENTITY STATEMENT IS REQUIRED (37 C.F.R. § 1.27), EXCEPT  
IF ONE FILED IN A PRIOR APPLICATION IS RELIED UPON (37 C.F.R. § 1.28)**ADDRESS TO: Assistant Commissioner for Patents  
Box Patent Application  
Washington, DC 20231

5. ☐ Microfiche Computer Program (Appendix)
6. Nucleotide and/or Amino Acid Sequence Submission  
(if applicable, all necessary)
- a. ☐ Computer Readable Copy
  - b. ☐ Paper Copy (identical to computer copy)
  - c. ☐ Statement verifying identity of above copies

**ACCOMPANYING APPLICATION PARTS**

7. ☒ Assignment Papers (cover sheet & document(s))
8. ☐ 37 C.F.R. §3.73(b) Statement of Power of Attorney  
(when there is an assignee)
9. ☐ English Translation Document (if applicable)
10. ☐ Information Disclosure Statement (IDS)/PTO-1449 ☐ Copies of IDS Citations
11. ☐ Preliminary Amendment
12. ☒ Return Receipt Postcard (MPEP 503)  
(Should be specifically itemized)
13. ☐ \* Small Entity Statement(s) ☐ Statement filed in prior application,  
Status still proper and desired  
(PTO/SB/09-12)
14. ☒ Certified Copy of Priority Document(s)  
(if foreign priority is claimed)
15. ☐ Other: \_\_\_\_\_

16. If a CONTINUING APPLICATION, check appropriate box, and supply the requisite information below and in a preliminary amendment:

☐ Continuation ☐ Divisional ☐ Continuation-in-part (CIP) of prior application No: \_\_\_\_\_

Prior application information: Examiner \_\_\_\_\_ Group / Art Unit: \_\_\_\_\_

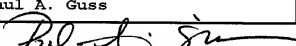
**For CONTINUATION or DIVISIONAL APPS only:** The entire disclosure of the prior application, from which an oath or declaration is supplied under Box 4b, is considered a part of the disclosure of the accompanying continuation or divisional application and is hereby incorporated by reference. The incorporation can only be relied upon when a portion has been inadvertently omitted from the submitted application parts.**17. CORRESPONDENCE ADDRESS**☒ Customer Number or Bar Code Label

22712

or ☒ Correspondence address below

(Insert Customer No. or Attach bar code label here)

Name	Paul A. Guss				
Address	775 South 23rd Street First Floor, Suite 2				
City	Arlington	State	VA	Zip Code	22202
Country		Telephone	(703) 486-2710	Fax	(703) 486-2720

Name (Print/Type)	Paul A. Guss	Registration No. (Attorney/Agent)	33,099
Signature		Date	Sept. 8, 2000

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Box Patent Application, Washington, DC 20231.

# FEE TRANSMITTAL

## for FY 2000

Patent fees are subject to annual revision.

Small Entity payments must be supported by a small entity statement, otherwise large entity fees must be paid. See Forms PTO/SB/09-12  
See 37 C.F.R. §§ 1.27 and 1.28

TOTAL AMOUNT OF PAYMENT (\$) 1120.00

### Complete if Known

Application Number	
Filing Date	September 8, 2000
First Named Inventor	Shinya MATSUMOTO et al.
Examiner Name	
Group / Art Unit	
Attorney Docket No.	CS-20-000908

### METHOD OF PAYMENT (check one)

1. ☒ The Commissioner is hereby authorized to charge indicated fees and credit any overpayments to:

Deposit Account Number 07-2519

Deposit Account Name Paul A. Guss

☒ Charge Any Additional Fee Required Under 37 CFR §§ 1.16 and 1.17

2. ☒ Payment Enclosed:

☒ Check ☐ Money Order ☐ Other

### FEE CALCULATION

#### BASIC FILING FEE

Large Entity Fee Code (\$)	Small Entity Fee Code (\$)	Fee Description	Fee Paid
01 690	201 345	Utility filing fee	690.00
06 310	206 155	Design filing fee	
107 480	207 240	Plant filing fee	
08 690	208 345	Reissue filing fee	
14 150	214 75	Provisional filing fee	

SUBTOTAL (1) (\$) 690.00

#### EXTRA CLAIM FEES

Total Claims	Extra Claims	Fee from below	Fee Paid
20	-20** = 0	18	
Independent Claims	8 - 3** = 5	78	390.00
Multiple Dependent			

\*\*or number previously paid, if greater; For Reissues, see below

Large Entity Fee Code (\$)	Small Entity Fee Code (\$)	Fee Description	Fee Paid
103 18	203 9	Claims in excess of 20	
102 78	202 39	Independent claims in excess of 3	
104 260	204 130	Multiple dependent claim, if not paid	
109 78	209 39	** Reissue independent claims over original patent	
110 18	210 9	** Reissue claims in excess of 20 and over original patent	

SUBTOTAL (2) (\$) 390.00

### FEE CALCULATION (continued)

#### 3. ADDITIONAL FEES

Large Entity Fee Code (\$)	Small Entity Fee Code (\$)	Fee Description	Fee Paid
105 130	205 65	Surcharge - late filing fee or oath	
127 50	227 25	Surcharge - late provisional filing fee or cover sheet	
139 130	139 130	Non-English specification	
147 2,520	147 2,520	For filing a request for reexamination	
112 920*	112 920*	Requesting publication of SIR prior to Examiner action	
113 1,840*	113 1,840*	Requesting publication of SIR after Examiner action	
115 110	215 55	Extension for reply within first month	
116 380	216 190	Extension for reply within second month	
117 870	217 435	Extension for reply within third month	
118 1,360	218 680	Extension for reply within fourth month	
128 1,850	228 925	Extension for reply within fifth month	
119 300	219 150	Notice of Appeal	
120 300	220 150	Filing a brief in support of an appeal	
121 260	221 130	Request for oral hearing	
138 1,510	138 1,510	Petition to institute a public use proceeding	
140 110	240 55	Petition to revive - unavoidable	
141 1,210	241 605	Petition to revive - unintentional	
142 1,210	242 605	Utility issue fee (or reissue)	
143 430	243 215	Design issue fee	
144 580	244 290	Plant issue fee	
122 130	122 130	Petitions to the Commissioner	
123 50	123 50	Petitions related to provisional applications	
126 240	126 240	Submission of Information Disclosure Stmt	
581 40	581 40	Recording each patent assignment per property (times number of properties)	40.00
146 690	246 345	Filing a submission after final rejection (37 CFR § 1.129(a))	
149 690	249 345	For each additional invention to be examined (37 CFR § 1.129(b))	
Other fee (specify)			
Other fee (specify)			

\* Reduced by Basic Filing Fee Paid

SUBTOTAL (3) (\$) 40.00

### SUBMITTED BY

Name (Print/Type)	Paul A. Guss	Registration No. (Attorney/Agent)	33,099	Telephone	703-486-2710
Signature			Date	09/08/2000	

### WARNING:

Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office, Washington, DC 20231  
DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Washington, DC 20231.

METHOD OF AND APPARATUS FOR RENDERING IMAGE,  
RECORDING MEDIUM, AND PROGRAM

BACKGROUND OF THE INVENTION

Field of the Invention:

The present invention relates to a method of and an apparatus for rendering an image representing a fluid motion such as a stream of water or a flow of smoke, a recording medium which stores a program and data for carrying out such an image rendering process, and a program for carrying out such an image rendering process.

Description of the Related Art:

Recently, various computer graphics (CG) processing techniques including hidden line processing, hidden surface removal, smooth shading, texture mapping, etc. have been in rapid progress in combination with quickly growing hardware technologies.

According to a general CG processing scheme, a plurality of three-dimensional shapes (objects) are generated by three-dimensional modeling of CAD, and a rendering process is performed by applying colors and shades to the objects, adding optical properties including mirror reflection, diffuse reflection, refraction, transparency, etc. to the objects, adding surface patterns to the objects, and plotting images depending on surroundings such as window and scenery reflections and ambient light.

For rendering a fluid motion such as a stream of water

or a flow of smoke, for example, it has been attempted to move a number of objects at random and thereafter map an image of water or smoke onto the objects.

Specifically, for processing ordinary polygons, a geometry operation unit performs perspective projection on the vertexes of polygons, and a rendering processor stores the obtained polygons in a display rendering memory.

For example, if a fluid motion is to be rendered by moving  $n$  objects at random and then mapping a texture onto the objects, then it is necessary to perform the above process, i.e., the ordinary polygon processing sequence with respect to an object,  $n$  times, and, as a result, it is necessary to perform ( $n \times$  the number of polygon vertexes) perspective projection processes, and also to perform ( $n \times$  the number of polygons) rendering processes.

In order to render a stream of water or a flow of smoke realistically, therefore, a number of objects need to be prepared, and the period of time required to compute and render those objects increases in proportion to the number of objects involved. Consequently, those objects are displayed on a display monitor at a reduced speed.

#### SUMMARY OF THE INVENTION

It is therefore an object of the present invention to provide a method of and an apparatus for rendering an image by moving a texture simulatively on at least one object thereby to greatly reduce the period of time required to

calculate and render a fluid motion and also to render a high-quality image, a recording medium which stores a program and data for carrying out such an image rendering process, and a program for carrying out such an image rendering process.

According to the present invention, a texture is moved simulatively on at least one object thereby to render an image.

Specifically, one or more semitransparent textures are placed on the surface of at least one semitransparent or transparent object, and the one or more semitransparent textures are moved simulatively in an arbitrary direction, thereby rendering an image. The semitransparent or transparent object may be arranged in one or multiple layers.

Texture images that are moved simulatively on the surface of the object are superposed on the object, thereby generating or animating an infinite number of random patterns to express a fluid motion such as a stream of water or a flow of smoke.

Alternatively, one or more semitransparent textures may be placed on the surface of a plurality of semitransparent or transparent polygons which make up the object, and the one or more semitransparent textures may be moved simulatively in an arbitrary direction, thereby rendering an image. The semitransparent or transparent polygons may be arranged in one or multiple layers.

Texture images that are moved simulatively on the sur-

face of the polygons are superposed on the polygons, thereby generating or animating an infinite number of random patterns to express a fluid motion such as a stream of water or a flow of smoke.

5           Since an image is rendered by moving a texture simultaneously on at least one object, the period of time required to calculate and render a fluid motion can greatly be reduced, and a high-quality image can be rendered.

10           The above and other objects, features, and advantages of the present invention will become more apparent from the following description when taken in conjunction with the accompanying drawings in which a preferred embodiment of the present invention is shown by way of illustrative example.

#### 15           BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram of an entertainment apparatus according to an embodiment of the present invention;

20           FIG. 2 is a perspective view showing the manner in which the water of a river hits a post and flows in divided streams on both sides of the post;

FIG. 3 is a diagram showing the object illustrated in FIG. 2 as it is divided into a number of polygons;

FIGS. 4A and 4B are views of texture images;

25           FIG. 5 is a diagram illustrative of the directions in which texture images stored in a texture rendering area move;

FIG. 6 is a view showing objects that are arranged in

superposed layers;

FIG. 7A is a view showing the manner in which water pouring from a faucet is stored in a transparent water tank;

FIG. 7B is a view showing the image illustrated in FIG. 7A as it is separated into a plurality of objects;

FIG. 8 is a functional block diagram of a rendering means according to the embodiment of the present invention;

FIG. 9 is a diagram showing details of an object information table;

FIG. 10 is a diagram showing details of a texture information table;

FIG. 11 is a diagram showing details of a polygon information table;

FIG. 12 is a diagram showing details of a movement information table;

FIGS. 13 and 14 are a flowchart of a processing sequence of the rendering means;

FIG. 15 is a flowchart of a processing sequence of a texture rendering means;

FIG. 16 is a flowchart of a processing sequence of a rendering means;

FIG. 17 is a flowchart of a processing sequence of a texture moving means; and

FIG. 18 is a flowchart of a processing sequence of a texture mapping means.

#### DESCRIPTION OF THE PREFERRED EMBODIMENT

An embodiment in which a method of and an apparatus for rendering an image according to the present invention are applied to an entertainment apparatus for performing three-dimensional CG processing, and a recording medium and a program according to the present invention are applied to a recording medium storing a program and data executed by the entertainment apparatus and such a program, will be described below with reference to FIGS. 1 through 18.

As shown in FIG. 1, an entertainment apparatus 10 comprises a main CPU 12 for controlling the entertainment apparatus 10, a main memory 14 for storing various programs to be run and various data, an image processor 18 for generating image data under the control of the main CPU 12 and outputting the generated image data to a display monitor 16, e.g., a CRT, and an input/output port 20 for sending data to and receiving data from external devices.

The main memory 14, the image processor 18, and the input/output port 20 are connected to the main CPU 12 via a bus 22. To the input/output port 20, there are connected an input device 24 for inputting data (key entry data, coordinate data, etc.) to the entertainment apparatus 10, and an optical disk drive 26 for playing back an optical disk such as a CD-ROM or the like in which various programs and data (object-related data, texture data, etc.) are stored.

The image processor 18 comprises a rendering engine 30, a memory interface 32, an image memory 34, and a display controller 36 such as a programmable CRT controller or the



like.

The rendering engine 30 serves to render image and store image data in the image memory 34 via the memory interface 32 based on a rendering command supplied from the main CPU 12.

A first bus 38 is connected between the memory interface 32 and the rendering engine 30, and a second bus 40 is connected between the memory interface 32 and the image memory 34. Each of the first and second buses 38, 40 has a 128-bit width, for example, for allowing the rendering engine 30 to render and store image data in the image memory 34 at a high speed.

The rendering engine 30 is capable of rendering image data of  $320 \times 240$  pixels or image data of  $640 \times 480$  pixels according to the NTSC or PAL system on a real-time fashion, i.e., more than ten times to several ten times in  $1/60$  seconds to  $1/30$  seconds.

The image memory 34 is of a unified memory structure that is able to designate a texture rendering area 34a and a display rendering area 34b as the same area (see FIG. 8).

The image controller 36 writes texture data read from the optical disk via the optical disk drive 26 or texture data generated in the main memory 14 via the memory interface 32 into the texture rendering area 34a of the image memory 34, and reads image data stored in the display rendering area 34b of the image memory 34 via the memory interface 32 and outputs the read image data to the display moni-

tor 16 to display an image on its display screen.

A characteristic function of the entertainment apparatus 10 will be described below with reference to FIGS. 2 through 18.

According to the characteristic function, an image is rendered by moving a texture simulatively on at least one object.

Specifically, as shown in FIG. 2, it is assumed that the water of a river 200 hits a post 202 standing in the river 200 and flows in divided streams on both sides of the post 202.

As shown in FIG. 3, an object 204 representing the river 200 except for the post 202 is constructed of a number of semitransparent or transparent polygons 206, and a semitransparent texture image 208 (see FIG. 4A, for example) is mapped onto each of the polygons 206. The semitransparent texture image 208 may be an image expressing diffused reflections of light from waves, such as images containing randomly scattered semitransparent spots, as shown in FIGS. 4A and 4B.

The texture image 208 to be used is stored in the texture rendering area 34a of the image memory 34. Thereafter, required polygons 206 are stored in the display rendering area 34b of the image memory 34 and the texture image 208 is mapped onto each of the polygons 206, after which the polygons 206 with the mapped texture image 208 are displayed.

Then, the texture image 208 in the texture rendering

area 34a is moved a predetermined interval in an arbitrary direction and restored in the texture rendering area 34a, and the restored texture image 208 is mapped onto each of the polygons 206, after which the polygons 206 with the mapped texture image 208 are displayed.

For example, an arbitrary texture image 208 is stored in the texture rendering area 34a in a pattern corresponding to the shape of the object 204 that is made up of the polygons 206.

Specifically, texture images 208 covering five polygons 206 in a horizontal succession indicated by ① in FIG. 3 are stored in an area ① in the texture rendering area 34a as shown in FIG. 5, and texture images 208 covering five polygons 206 in a horizontal succession indicated by ② in FIG. 3 are stored in an area ② in the texture rendering area 34a as shown in FIG. 5.

Then, a texture image 208 corresponding to one polygon 206 indicated by ③ in FIG. 3 is stored in an area ③ in the texture rendering area 34a as shown in FIG. 5, and texture images 208 covering three polygons 206 in a horizontal succession indicated by ④ in FIG. 3 are stored in an area ④ in the texture rendering area 34a as shown in FIG. 5. Similarly, texture images 208 covering remaining polygons indicated by ⑤ - ⑧ in FIG. 3 are stored in corresponding areas in the texture rendering area 34a as shown in FIG. 5.

Then, a texture image 208 indicated by (1) in FIG. 5 is mapped onto a polygon 206 indicated by (1) in FIG. 3. Like-

wise, other texture images 208 stored in the texture rendering area 34a are mapped onto corresponding polygons 206. When all the stored texture images 208 are mapped onto the corresponding polygons 206, an initial image is displayed.

5           Thereafter, as shown in FIG. 5, each of the texture images 208 is moved a predetermined interval in the direction indicated by the arrow, and restored in the texture rendering area 34a. Some texture images 208, e.g., those indicated by ①, ② in FIG. 5, are cycled to express an image in a circulating fashion. Specifically, when a succession of texture images 208 are moved a given interval, a texture image 208 is pushed out of the end of the succession, i.e., erased from the end of the texture rendering area 34a, and put back to the beginning of the succession, i.e., restored in the beginning of the texture rendering area 34a.

10           Each of the texture images 208 indicated by (6), (12), (16), (19) in FIG. 5 is moved in the two directions indicated by the arrows, and restored in the texture rendering area 34a.

15           In this manner, as shown in FIG. 2, it is possible to display a realistic image of the flowing water of the river 200, showing the water of the river 200 flowing in one direction and changing its direction around the post 202.

20           In the above example, the characteristic function of the present invention is applied to a single object 204. However, the characteristic function of the present invention is also applicable to a number of objects arranged in

superposed layers.

For example, as shown in FIG. 6, two objects, i.e., first and second objects, 204a, 204b are prepared for the river 200 with the post 202 standing therein. Each of the first and second objects 204a, 204b is made up of a number of polygons 206. A first texture image 208a is mapped onto each of the polygons 206 of the first object 204a, and a second texture image 208b is mapped onto each of the polygons 206 of the second object 204b.

When the first and second objects 204a, 204b are arranged in superposed layers, the first texture image 208a and the second texture image 208b are superposed, producing various patterns, as shown in an encircled area A in FIG. 6.

The first and second texture images 208a, 208b are moved on the first and second objects 204a, 204b with time in an arbitrary direction and restored in the texture rendering area 34a. Therefore, an infinite number of random patterns are generated or animated on the first and second objects 204a, 204b superposed in layers, making it possible to express a fluid motion such as a stream of water or a flow of smoke in a realistic manner.

As shown in FIG. 7A, one example of a number of objects arranged in superposed layers is a scene in which water pouring from a faucet 210 is stored in a transparent water tank 212. In this example, as shown in FIG. 7B, an object 214 represents the manner in which water flows out from the faucet 210, two objects 216, 218 represent the manner in

which water from the faucet 210 spreads out on the water surface in the water tank 212, and two objects 220, 222 represent the manner in which water is stored in the water tank 212. These objects 214, 216, 218, 220, 222 are arranged in superposed layers.

Each of the objects 214, 216, 218, 220, 222 is constructed of a number of polygons. Individual texture images 208 stored in the texture rendering area 34a are mapped onto the respective polygons, and moved on the objects 214, 216, 218, 220, 222 with time in an arbitrary direction and restored in the texture rendering area 34a. As a result, it is possible to display a realistic image of water pouring from the faucet 210 and stored in the transparent water tank 212.

An example of software, i.e., a rendering means 300 (see FIG. 8), to perform the above function will be described below with reference to FIGS. 8 through 18.

The rendering means 300 is supplied to the entertainment apparatus 10 from a randomly accessible recording medium such as a CD-ROM or a memory card or via a network. It is assumed here that the rendering means 300 is read into the entertainment apparatus 10 from an optical disk 42 such as a CD-ROM.

The rendering means 300 is downloaded in advance from the optical disk 42 played back by the entertainment apparatus 10 into the main memory 14 of the entertainment apparatus 10 according to a predetermined process, and executed by

the main CPU 12.

As shown in FIG. 8, the rendering means 300 comprises a table reading means 302 for reading various tables including an object information table 322, a plurality of texture information tables 304, a plurality of polygon information tables 308, and a plurality of movement information tables 312 which are recorded in the optical disk 42, a texture rendering means 306 for rendering and storing necessary texture images 208 in the texture rendering area 34a of the image memory 34 based on the content of one of the texture information tables 304, an image rendering means 310 for rendering and storing polygons 206 in the display rendering area 34b of the image memory 34 based on the texture images 208 stored in the texture rendering area 34a and the content of one of the polygon information tables 308, and mapping necessary texture images 208 onto the polygons 206, a texture moving means 314 for moving the texture images 208 stored in the texture rendering area 34a based on the content of one of the movement information tables 312 and restoring the moved texture images 208 in the texture rendering area 34a, a texture mapping means 316 for mapping moved texture images 208 onto the polygons 206 stored in the display rendering area 34b, an end determining means 318 for determining whether the processing sequence of the rendering means 300 is completed or not, and a display processing means 320 for outputting image data stored in the display rendering area 34b to the display monitor 16 and displaying an image on the

display monitor 16 based on the image data.

The various tables used by the rendering means will be described below with reference to FIGS. 9 through 12.

As shown in FIG. 9, the object information table 322 stores a plurality of records which correspond to respective objects, each containing the number of polygons 206 making up an object 204, the address of a corresponding texture information table 304, the address of a corresponding polygon information table 308, and the address of a corresponding movement information table 312.

As shown in FIG. 10, the texture information table 304 stores a plurality of records which correspond to respective textures, each containing corresponding record indication information in a corresponding texture table, which stores leading addresses of stored texture image data in respective records, data length of a texture image to be used, and a rendering range (coordinates) for storing the texture image in the texture rendering area 34a.

As shown in FIG. 11, the polygon information table 308 stores a plurality of records which correspond to respective polygons, each containing the vertex coordinates of a polygon 206, and the rendering range (coordinates) for storing the texture image 208 to be used in the texture rendering area 34a.

As shown in FIG. 12, the movement information table 312 stores a plurality of records which correspond to respective texture images 208 to be moved, each containing a range (co-



ordinates) for storing a texture image 208 to be moved in the texture rendering area 34a, direction information as to the direction (up, down, left, or right) in which to move the texture image 208, and circulation information as to whether the texture image 208 is to be circulated or not.

The processing sequence of the rendering means 300 will be described below with reference to FIGS. 13 through 18.

In step S1 shown in FIG. 13, the rendering means 300 stores an initial value "0" into an index register i used to retrieve objects 204, thereby initializing the index register i.

In step S2, the table reading means 302 reads an ith record from the object information table 322.

In step S3, the end determining means 318 decides whether the content of the read ith record is an EOF (End-Of-File) code indicative of the end of the object information table 322 or not.

If the content of the read ith record is not an EOF code, then control goes to step S4 in which the table reading means 302 reads the address of the corresponding texture information table 304 and the address of the corresponding polygon information table 308 from the ith record in the object information table 322, and reads the corresponding texture information table 304 and the corresponding polygon information table 308.

In step S5, the texture rendering means 306 performs its own processing sequence, which is illustrated in FIG.

15. In step S101 shown in FIG. 15, the texture rendering means 306 stores an initial value "0" into an index register j used to retrieve texture images 208, thereby initializing the index register j.

5 In step S102, the texture rendering means 306 reads a jth record from the corresponding texture information table 304.

10 In step S103, the texture rendering means 306 decides whether the content of the read jth record is an EOF code indicative of the end of the texture information table 304 or not.

15 If the content of the read jth record is not an EOF code, then control goes to step S104 in which the texture rendering means 306 reads texture image data based on the record indication information in the texture table and the data length of a texture image 208 stored in the jth record, and stores the texture image data in the portion of the texture rendering area 34a which corresponds to the rendering range that is stored in the jth record.

20 In step S105, the texture rendering means 306 increments the value of the index register j by "1". Then, control goes back to step S102 to repeat the above process based on the content of a next record in the texture information table 304.

25 If the content of the read jth record is an EOF code indicative of the end of the texture information table 304, then the processing sequence of the texture rendering means

306 is ended.

Control returns to the main routine shown in FIG. 13. In step S6, the image rendering means 310 performs its own processing sequence, which is illustrated in FIG. 16. In  
5 step S201 shown in FIG. 16, the image rendering means 310 stores an initial value "0" into an index register k used to retrieve polygons 206, thereby initializing the index register k.

10 In step S202, the image rendering means 310 reads a kth record from the corresponding polygon information table 308.

In step S203, the image rendering means 310 stores a kth polygon 206 into the portion of the display rendering area 34b which corresponds to the rendering range derived from the vertex coordinates.

15 In step S204, the image rendering means 310 reads a texture image 208 from the rendering range for the texture image 208 to be used, among the texture images 208 stored in the texture rendering area 34a, and maps the read texture image 208 onto the kth polygon 206.

20 In step S205, the image rendering means 310 increments the value of the index register k by "1". Thereafter, in step S206, the image rendering means 310 decides whether all the polygons 206 have been processed or not, based on whether or not the value of the index register k is equal to or  
25 greater than the number M of polygons.

If not all the polygons 206 have been processed, then control returns to step S202 for rendering a next polygon

206 and mapping a necessary texture image 208 onto the next polygon 206.

If all the polygons 206 have been processed in step S206, then the processing sequence of the image rendering means 310 comes to an end.

Control returns to the main routine shown in FIG. 13. In step S7, the rendering means 300 increments the value of the index register i by "1". Then, control goes back to step S2 for repeating the above process with respect to a next object 204.

If the content of the read ith record is an EOF code indicating the end of the object information table 322 in step S3, then control jumps to step S8 in which the display processing means 320 outputs the image data stored in the display rendering area 34b to the display monitor 16, which displays an image based on the supplied image data.

In step S9 shown in FIG. 14, the rendering means 300 stores an initial value "0" into the index register i used to retrieve objects 204, thereby initializing the index register i.

In step S10, the table reading means 302 reads an ith record from the object information table 322.

In step S11, the end determining means 318 decides whether the content of the read ith record is an EOF code indicative of the end of the object information table 322 or not.

If the content of the read ith record is not an EOF

code, then control goes to step S12 in which the table reading means 302 reads the address of the corresponding polygon information table 308 and the address of the corresponding movement information table 312 from the *i*th record in the object information table 322, and reads the corresponding polygon information table 308 and the corresponding movement information table 312.

In step S13, the texture moving means 314 performs its own processing sequence, which is illustrated in FIG 17. In step S301 shown in FIG. 17, the texture moving means 314 stores an initial value "0" into an index register *m* used to retrieve movement information, thereby initializing the index register *m*.

In step S302, the texture moving means 314 reads an *m*th record from the movement information table 312.

In step S303, the texture moving means 314 decides whether the content of the read *m*th record is an EOF code indicative of the end of the movement information table 312 or not.

If the content of the read *m*th record is not an EOF code, then control goes to step S304 in which the texture moving means 314 moves a texture image 208 to be moved which corresponds to the rendering range for the texture image 208 stored in the *m*th record, among the texture images 208 stored in the texture rendering area 34a, by a predetermined interval in the direction information stored in the *m*th record, and restores the moved texture image 208 in the texture

rendering area 34a.

In step S305, the texture moving means 314 decides whether the texture image 208 needs to be circulated or not, based on the circulation information stored in the mth record.

If the texture image 208 needs to be circulated, then control goes to step S306 in which the texture moving means 314 restores any texture image 208 pushed out or erased from the end of the texture rendering area 34a due to the movement carried out in step S304, back in the beginning of the texture rendering area 34a.

After the processing in step S306 or if the texture image 208 does not need to be circulated in step S305, then control goes to step S307 in which the texture moving means 314 increments the value of the index register m by "1". Thereafter, control goes back to step S302 for moving and restoring the corresponding texture image 208 based on next movement information.

If the content of the read mth record is an EOF code indicative of the end of the movement information table 312 in step S303, then the processing sequence of the texture moving means 314 is ended.

Control returns to the main routine shown in FIG. 14. In step S14, the texture mapping means 316 performs its own processing sequence, which is illustrated in FIG. 18.

In step S401 shown in FIG. 18, the texture mapping means 316 stores an initial value "0" into the index regis-

ter k used to retrieve polygons 206, thereby initializing the index register k.

In step S402, the texture mapping means 316 reads a kth record from the corresponding polygon information table 308.

5 In step S403, the texture mapping means 316 reads a texture image 208 from the rendering range for the texture image 208 to be used, among the texture images 208 stored in the texture rendering area 34a, and maps the read texture image 208 onto the kth polygon 206.

10 In step S404, the texture mapping means 316 increments the value of the index register k by "1". Thereafter, in step S405, the texture mapping means 316 decides whether all the polygons 206 have been processed or not, based on whether or not the value of the index register k is equal to or  
15 greater than the number M of polygons.

If not all the polygons 206 have been processed, then control returns to step S402 for rendering a next polygon 206 and mapping a necessary texture image 208 onto the next polygon 206.

20 If all the polygons 206 have been processed in step S405, then the processing sequence of the texture mapping means 316 comes to an end.

Control returns to the main routine shown in FIG. 14.

In step S15, the rendering means 300 increments the value of the index register i by "1". Then, control goes back to step S10 for repeating the above process, i.e., the process of moving the texture image 208 and mapping the texture im-  
25

age 208 onto the polygon 206, with respect to a next object 204.

5 If the content of the read ith record is an EOF code indicating the end of the object information table 322 in step S11, then control jumps to step S16 in which the display processing means 320 outputs the image data stored in the display rendering area 34b to the display monitor 16, which displays an image based on the supplied image data.

10 In step S17, the end determining means 318 decides whether there is a program end request to finish the processing sequence of the rendering means 300 or not. If there is no program end request, then control goes back to step S9 for performing the above process, i.e., the process of moving the texture image 208 and mapping the texture image 208 onto the polygon 206, from the first object 204.

15 If there is a program end request in step S17, then the processing sequence of the rendering means 300 is finished.

20 As described above, the rendering means 300 according to the present invention renders an image by moving a texture image 208 simulatively on at least one object 204. Specifically, one or more semitransparent texture images 208 are placed on the surface of one or more semitransparent or transparent objects 204, and moved simulatively in an arbitrary direction, thereby rendering an image.

25 The texture images 208 that are moved simulatively on the surface of the objects 204 are superposed on the objects 204, thereby generating or animating an infinite number of



random patterns to express a fluid motion such as a stream of water or a flow of smoke.

Since the rendering means 300 renders an image by moving a texture image 208 simulatively on at least one object 204, the period of time required to calculate and render a fluid motion can greatly be reduced, and a high-quality image can be rendered.

Although a certain preferred embodiment of the present invention has been shown and described in detail, it should be understood that various changes and modifications may be made therein without departing from the scope of the appended claims.

What is claimed is:

1. A method of rendering an image, comprising the step  
of:

5 moving a texture simulatively on at least one object  
thereby to render an image.

2. A method according to claim 1, wherein said moving  
step comprises the steps of:

10 placing one or more semitransparent textures on the  
surface of at least one semitransparent or transparent ob-  
ject; and

15 moving said one or more semitransparent textures simu-  
latively in an arbitrary direction, thereby rendering an im-  
age.

3. A method according to claim 2, wherein said moving  
step further comprises the step of arranging said semitrans-  
parent or transparent object in one or multiple layers.

20 4. A method according to claim 1, wherein said moving  
step comprises the steps of:

25 placing one or more semitransparent textures on the  
surface of a plurality of semitransparent or transparent  
polygons which make up said object; and

moving said one or more semitransparent textures simu-  
latively in an arbitrary direction, thereby rendering an im-

age.

5 5. A method according to claim 4, wherein said moving step further comprises the step of arranging said one or more semitransparent or transparent polygons in one or multiple layers.

6. A method of processing an image, comprising the steps of:

10 storing a texture image in a texture rendering area of an image memory;

storing a polygon in a display rendering area of said image memory based on at least said texture image, and mapping the texture image onto said polygon;

15 moving the texture image stored in said texture rendering area in an arbitrary direction and restoring the moved texture image in said texture rendering area; and

mapping the moved texture image onto the polygon stored in said display rendering area.

20 7. An apparatus for rendering an image, comprising: rendering means for moving a texture simulatively on at least one object thereby to render an image.

25 8. An apparatus according to claim 7, wherein said rendering means comprises:

means for placing one or more semitransparent textures

on the surface of at least one semitransparent or transparent object, and moving said one or more semitransparent textures simulatively in an arbitrary direction, thereby rendering an image.

5

9. An apparatus according to claim 8, wherein said rendering means comprises:

object setting means for arranging said semitransparent or transparent object in one or multiple layers.

10. An apparatus according to claim 7, wherein said rendering means comprises:

means for placing one or more semitransparent textures on the surface of a plurality of semitransparent or transparent polygons which make up said object, and moving said one or more semitransparent textures simulatively in an arbitrary direction, thereby rendering an image.

11. An apparatus according to claim 10, wherein said rendering means comprises:

object setting means for arranging one or more semitransparent or transparent polygons in one or multiple layers.

12. An apparatus for processing an image, comprising:  
texture rendering means for storing a texture image in a texture rendering area of an image memory;

image rendering means for storing a polygon in a display rendering area of said image memory based on at least said texture image, and mapping the texture image onto said polygon;

5 texture moving means for moving the texture image stored in said texture rendering area in an arbitrary direction and restoring the moved texture image in said texture rendering area; and

10 texture mapping means for mapping the moved texture image onto the polygon stored in said display rendering area.

13. A recording medium storing a program and data, said program comprising the step of:

15 moving a texture simulatively on at least one object thereby to render an image.

14. A recording medium according to claim 13, wherein said moving step comprises the steps of:

20 placing one or more semitransparent textures on the surface of at least one semitransparent or transparent object; and

moving said one or more semitransparent textures simulatively in an arbitrary direction, thereby rendering an image.

25 15. A recording medium according to claim 14, wherein said moving step further comprises the step of:

arranging said semitransparent or transparent object in one or multiple layers.

16. A recording medium according to claim 15, wherein said moving step comprises the steps of:

placing one or more semitransparent textures on the surface of a plurality of semitransparent or transparent polygons which make up said object; and

moving said one or more semitransparent textures simul-  
latively in an arbitrary direction, thereby rendering an im-  
age.

17. A recording medium according to claim 16, wherein said moving step further comprises the step of:

arranging said semitransparent or transparent polygons which make up said object in one or multiple layers.

18. A recording medium storing a program and data, said program comprising the steps of:

storing a texture image in a texture rendering area of an image memory;

storing a polygon in a display rendering area of said image memory based on at least said texture image, and map-  
ping the texture image onto said polygon;

moving the texture image stored in said texture render-  
ing area in an arbitrary direction and restoring the moved  
texture image in said texture rendering area; and

mapping the moved texture image onto the polygon stored in said display rendering area.

19. A program which can be read and executed by a computer, comprising the step of:

moving a texture simulatively on at least one object thereby to render an image.

20. A program comprising the steps of:

storing a texture image in a texture rendering area of an image memory;

storing a polygon in a display rendering area of said image memory based on at least said texture image, and mapping the texture image onto said polygon;

moving the texture image stored in said texture rendering area in an arbitrary direction and restoring the moved texture image in said texture rendering area; and

mapping the moved texture image onto the polygon stored in said display rendering area.

## ABSTRACT OF THE DISCLOSURE

An apparatus for processing an image has a texture rendering unit for storing a texture image in a texture rendering area of an image memory, an image rendering unit for storing a polygon in a display rendering area of the image memory based on at least the texture image, and mapping the texture image onto the polygon, a texture moving unit for moving the texture image stored in the texture rendering area in an arbitrary direction and restoring the moved texture image in the texture rendering area, and a texture mapping unit for mapping the moved texture image onto the polygon stored in the display rendering area.



FIG. 1

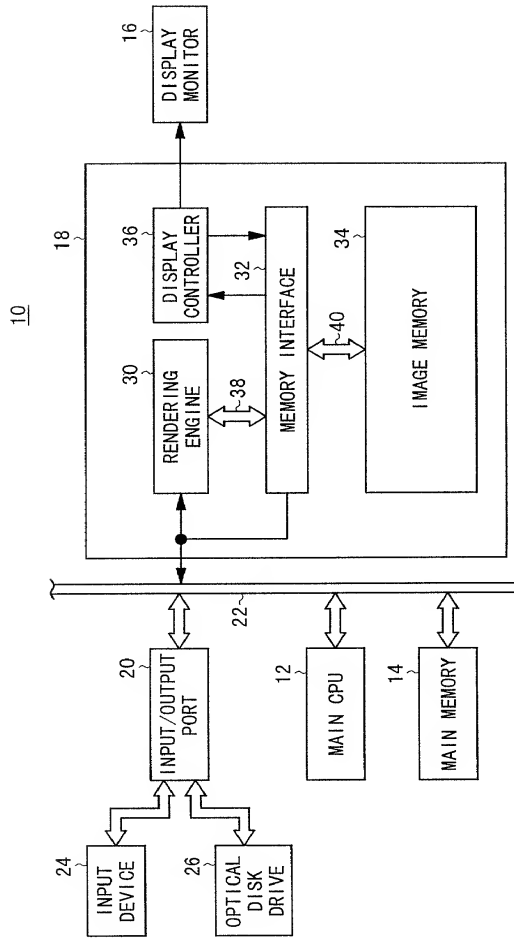
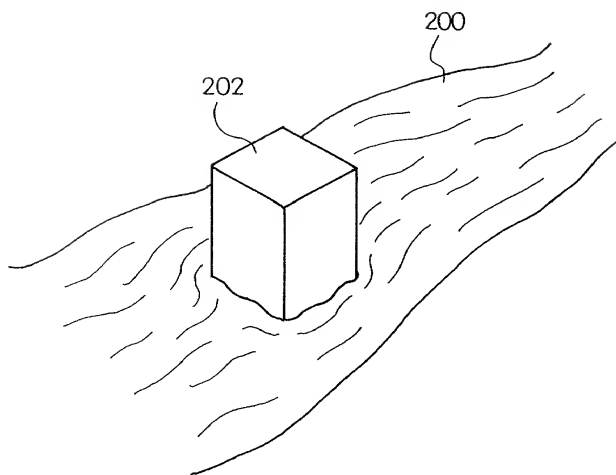


FIG. 2



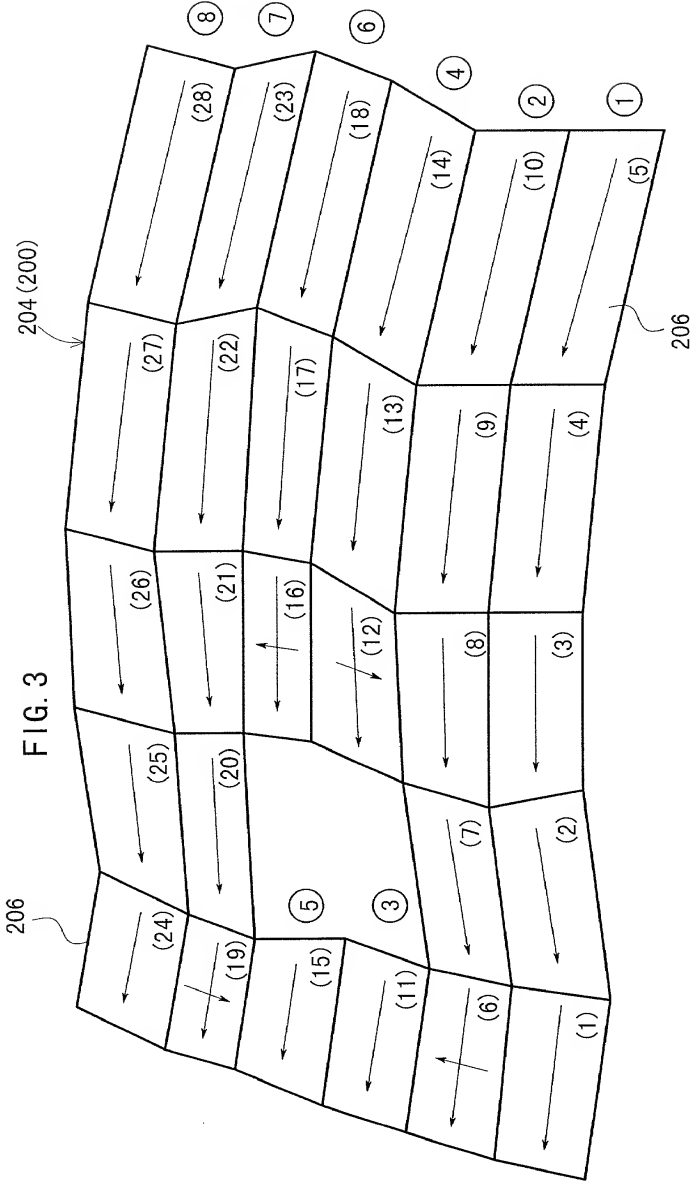


FIG. 4A

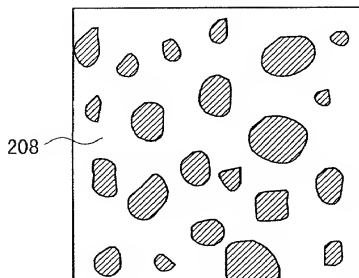


FIG. 4B

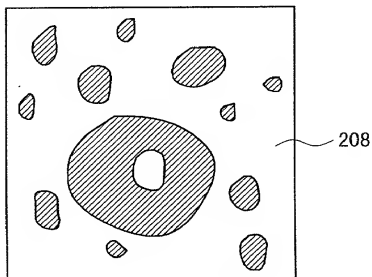


FIG. 5

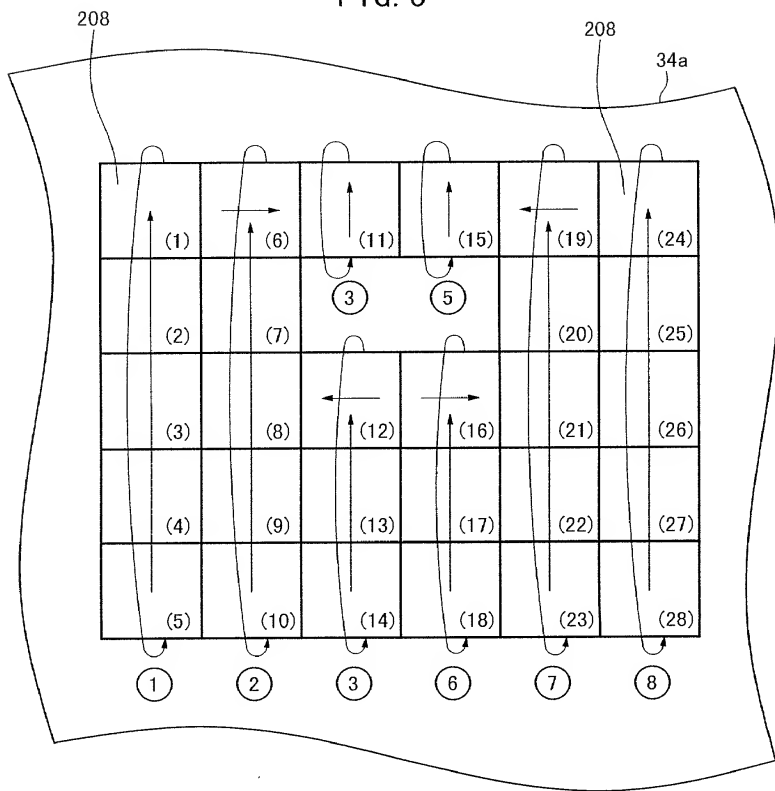


FIG. 6

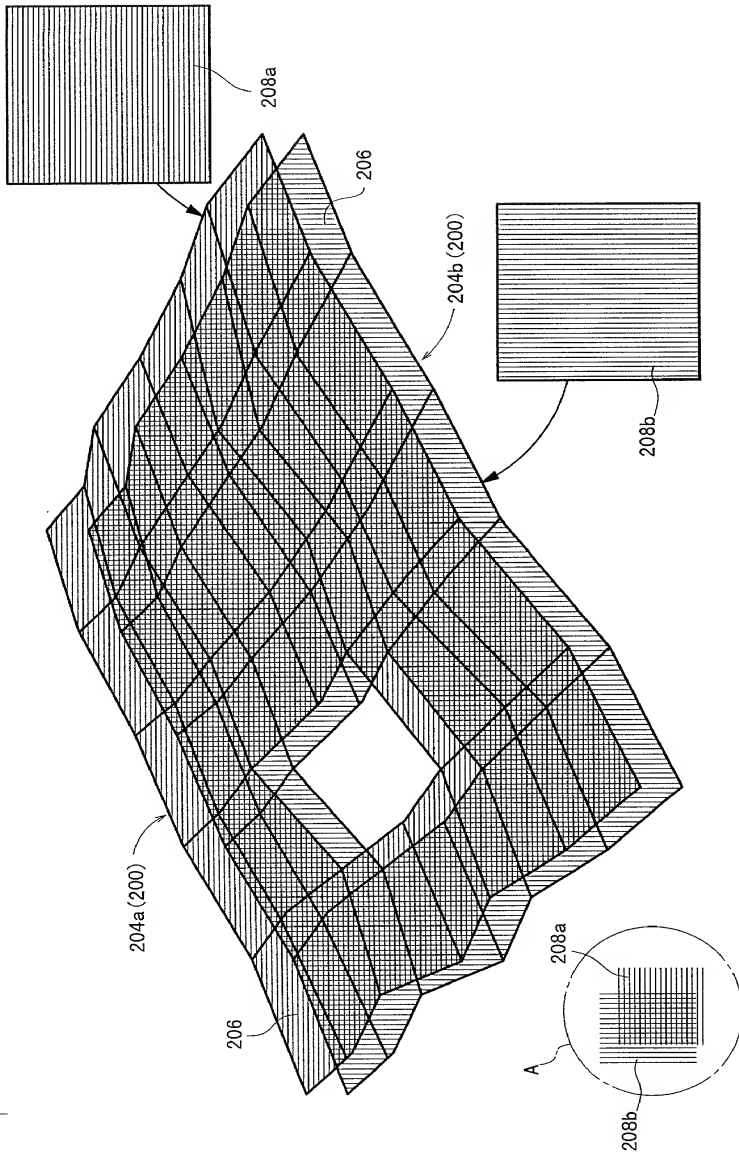


FIG. 7A

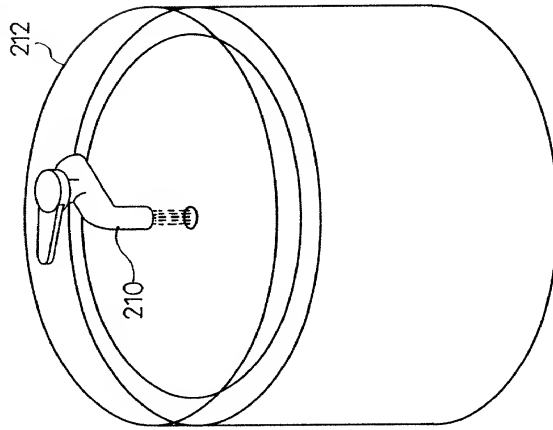


FIG. 7B

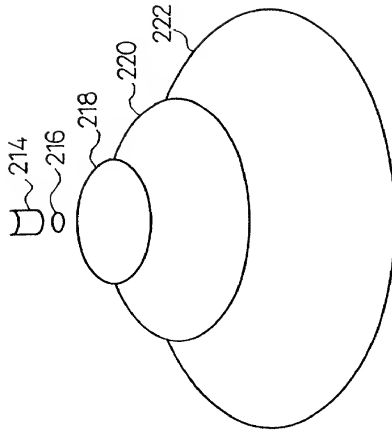
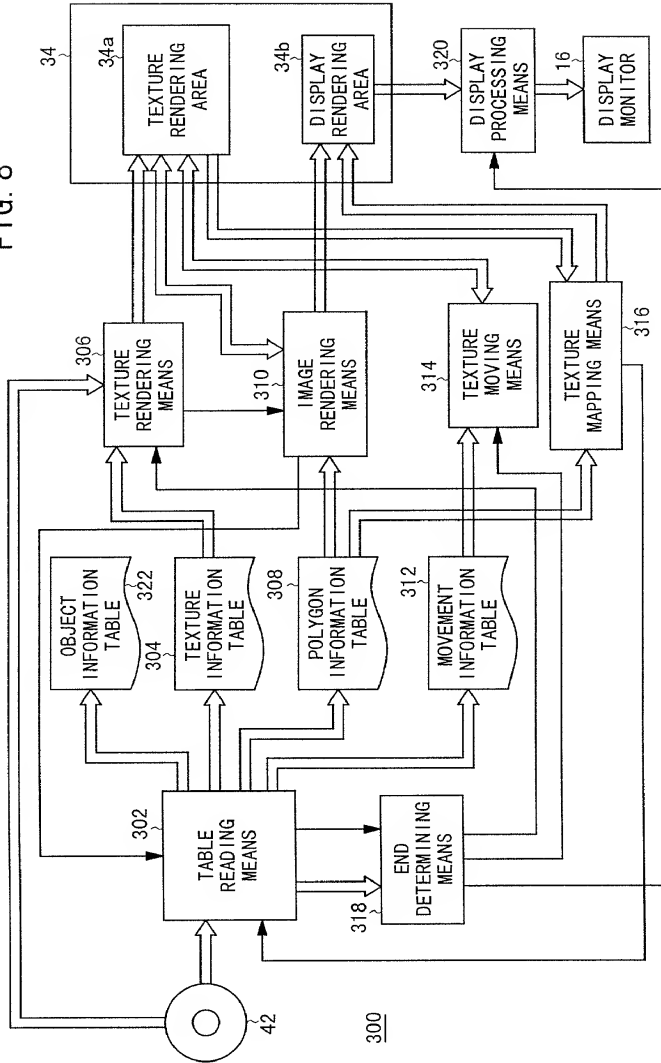


FIG. 8



300



FIG. 9

OBJECT INFORMATION TABLE (322)

RECORD 0	<ul style="list-style-type: none"><li>•NUMBER OF POLYGONS</li><li>•ADDRESS OF TEXTURE INFORMATION TABLE</li><li>•ADDRESS OF POLYGON INFORMATION TABLE</li><li>•ADDRESS OF MOVEMENT INFORMATION TABLE ETC.</li></ul>
RECORD 1	<ul style="list-style-type: none"><li>•NUMBER OF POLYGONS</li><li>•ADDRESS OF TEXTURE INFORMATION TABLE</li><li>•ADDRESS OF POLYGON INFORMATION TABLE</li><li>•ADDRESS OF MOVEMENT INFORMATION TABLE ETC.</li></ul>
	<ul style="list-style-type: none"><li>•</li><li>•</li><li>•</li></ul>

FIG. 10

TEXTURE INFORMATION TABLE (304)

RECORD 0	<div><div>•RECORD INDICATION INFORMATION OF TEXTURE TABLE</div><div>•DATA LENGTH OF TEXTURE IMAGE</div><div>•RENDERING RANGE (COORDINATES)</div><div>ETC.</div></div>
RECORD 1	<div><div>•RECORD INDICATION INFORMATION OF TEXTURE TABLE</div><div>•DATA LENGTH OF TEXTURE IMAGE</div><div>•RENDERING RANGE (COORDINATES)</div><div>ETC.</div></div>
	<div><div>•</div><div>•</div><div>•</div></div>

FIG. 11

POLYGON INFORMATION TABLE (308)

RECORD 0	<div>• VERTEX COORDINATES OF POLYGON</div> <div>• RENDERING RANGE (COORDINATES) FOR TEXTURE IMAGE TO BE USED ETC.</div>
RECORD 1	<div>• VERTEX COORDINATES OF POLYGON</div> <div>• RENDERING RANGE (COORDINATES) FOR TEXTURE IMAGE TO BE USED ETC.</div>
	<div>•</div> <div>•</div> <div>•</div>

FIG. 12

MOVEMENT INFORMATION TABLE (312)

RECORD 0	<ul style="list-style-type: none"><li>•RENDERING RANGE (COORDINATES) FOR TEXTURE IMAGE TO BE MOVED</li><li>•MOVEMENT DIRECTION</li><li>•CIRCULATION INFORMATION</li><li>ETC.</li></ul>
RECORD 1	<ul style="list-style-type: none"><li>•RENDERING RANGE (COORDINATES) FOR TEXTURE IMAGE TO BE MOVED</li><li>•MOVEMENT DIRECTION</li><li>•CIRCULATION INFORMATION</li><li>ETC.</li></ul>
	<div>• • •</div>

FIG. 13

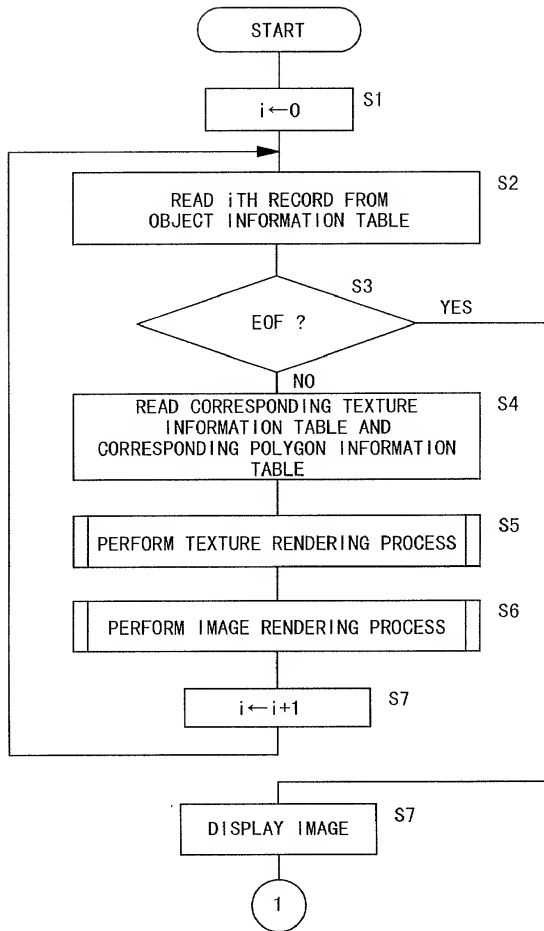


FIG. 14

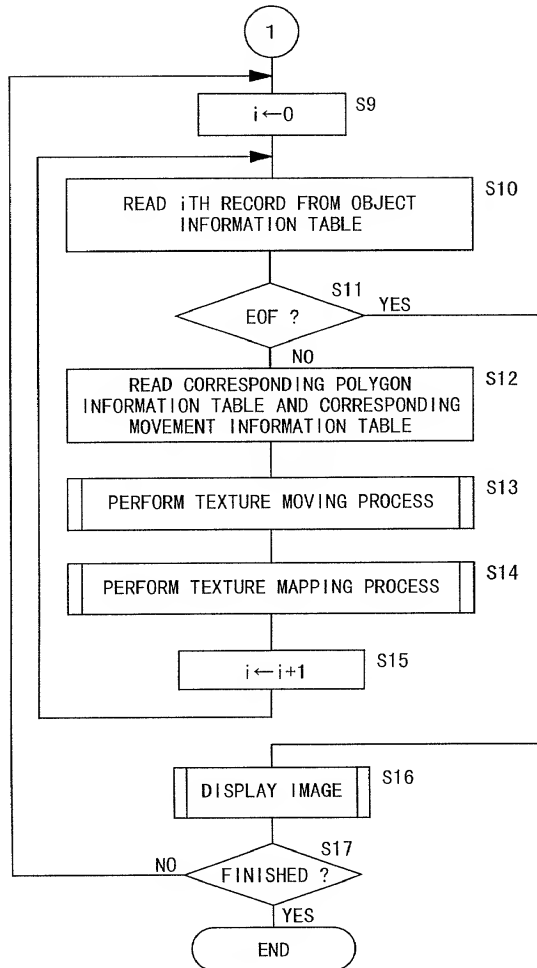


FIG. 15

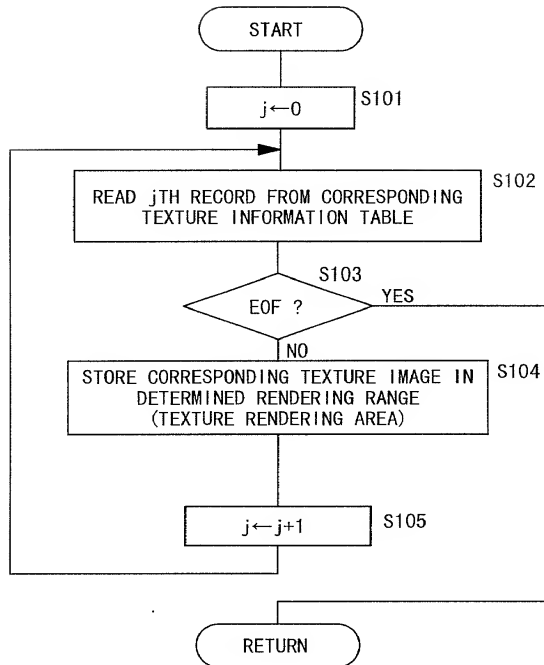


FIG. 16

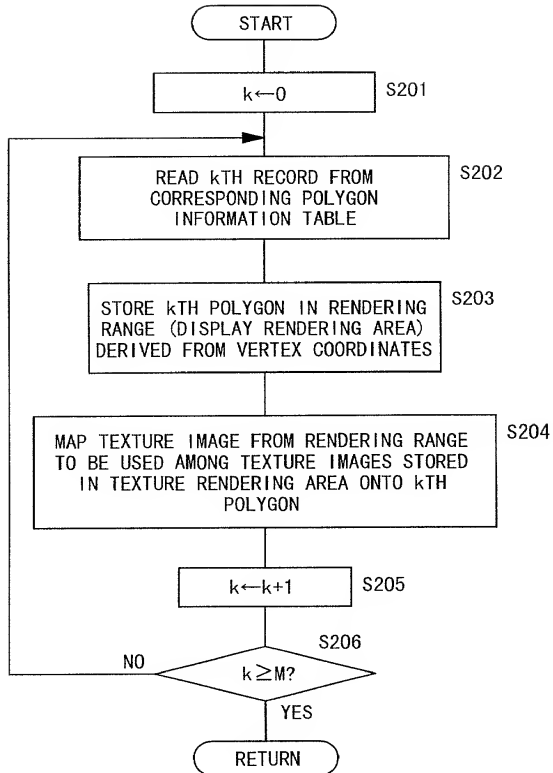




FIG. 17

## TEXTURE MOVING MEANS (314)

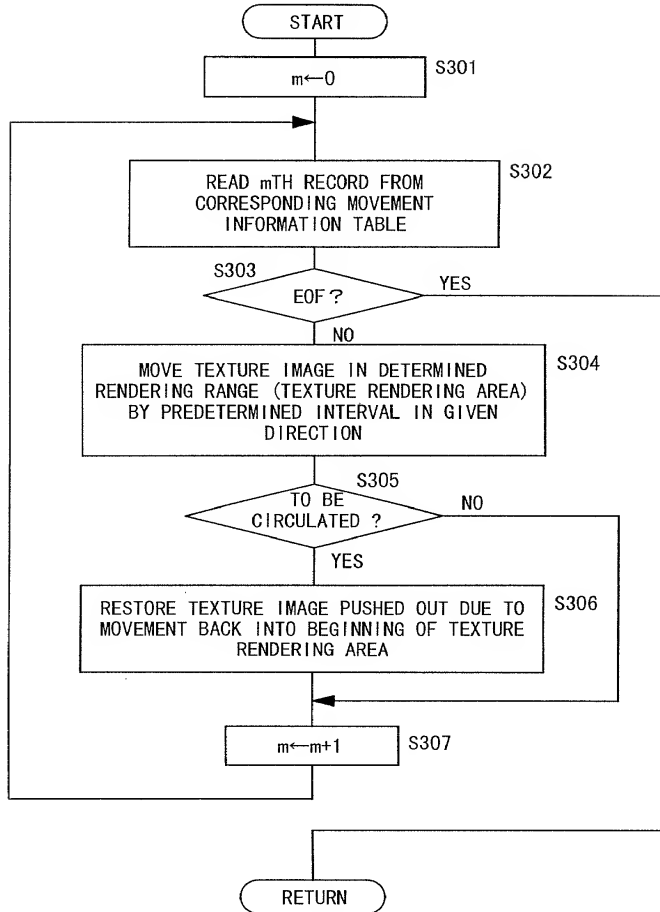
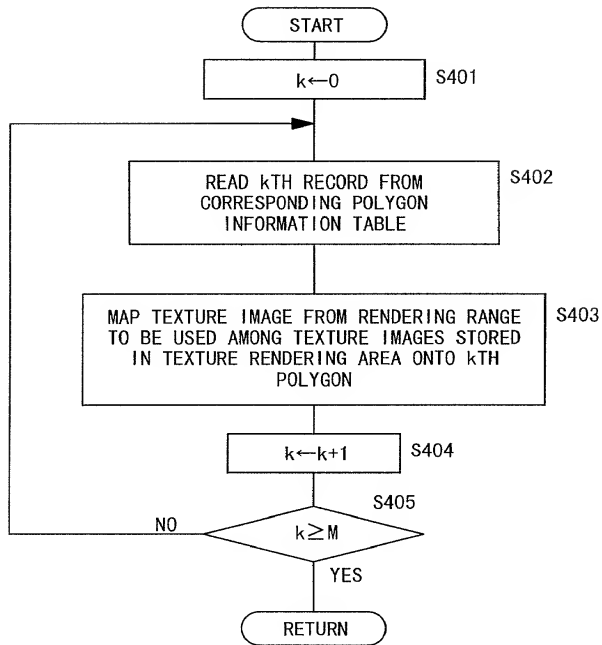


FIG. 18

TEXTURE MAPPING MEANS (316)



Declaration and Power of Attorney for Patent Application  
特許出願宣言書及び委任状

Dual English-Japanese Language Declaration  
英語－日本語併記宣言書

As a below named inventor, I hereby declare that:

下記の氏名の発明者として、私は以下の通り宣言します。

My residence, post office address and citizenship are as stated next to my name.

私の居所、郵便宛先、国籍は下記の私の氏名の後に記載された通りです。

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled.

下記の名称の発明に関して請求範囲に記載され、特許出願している発明内容について、私が最初かつ唯一の発明者（下記の氏名が一つの場合）もしくは最初かつ共同発明者（下記の氏名が複数の場合）であると信じています。

METHOD OF AND APPARATUS FOR RENDERING IMAGE, RECORDING MEDIUM, AND PROGRAM

the specification of which is attached hereto unless the following box is checked:

上記発明の明細書（下記の欄でX印がついていない場合は、本書に添付）は、

- ☐ was filed on \_\_\_\_\_ as United States Application Number or PCT International Application Number \_\_\_\_\_ and was amended on \_\_\_\_\_ (if applicable).
- ☐ \_\_\_\_\_ 年 \_\_\_\_\_ 月 \_\_\_\_\_ 日に提出され、米国出願番号または特許協定条約国際出願番号を \_\_\_\_\_ とし、（該当する場合）\_\_\_\_\_ に訂正されました。

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

私は、特許請求範囲を含む上記訂正後の明細書を検討し、内容を理解していることをここに表明します。

I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56.

私は、連邦規則法典第37編第1条56項に定義されるとおり、特許資格の有無について重要な情報を開示する義務があることを認めます。

I hereby claim foreign priority under Title 35, United States Code, Section 119 (a)-(d) or 365(b) of any foreign application(s) for patent or inventor's certificate, or Section 365(a) of any PCT International application which designated at least one country other than the United States, listed below and have also identified below, by checking the box, any foreign application for patent or inventor's certificate, or PCT International application having a filing date before that of the application on which priority is claimed.

私は、米国法典第35編119条(a)-(d)項又は365条(b)項に基づき下記の、米国以外の国の少なくとも一か国を指定している特許協力条約365条(a)項に基づく国際出願、又は外国での特許出願もしくは発明者証の出願についての外国優先権をここに主張するとともに、優先権を主張している本出願の前に出願された特許または発明者証の外国出願又は特許協力条約に基づく国際出願を以下に、枠内をマークすること、示しています。

Prior Foreign Application(s)

外国での先行出願

Priority Not Claimed

優先権主張なし

Patent Application  
No. 11-257933

Japan

10 September 1999

☐

(Number)

(Country)

(Day/Month/Year Filed)

(番号)

(国名)

(出願年月日)

☐

(Number)

(Country)

(Day/Month/Year Filed)

(番号)

(国名)

(出願年月日)

☐

(Number)

(Country)

(Day/Month/Year Filed)

(番号)

(国名)

(出願年月日)

I hereby claim the benefit under Title 35, United States Code, Section 119(e) of any United States provisional application(s) listed below.

私は、米国法典第35編119条(e)項に基づいて下記の米国特許出願規定に記載された権利をここに主張します。

(Application No.)

(出願番号)

(Filing Date)

(出願日)

(Application No.)

(出願番号)

(Filing Date)

(出願日)

I hereby claim the benefit under Title 35, United States Code, Section 120 of any United States application(s), or Section 365(c) of any PCT International application designating the United States, listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States or

PCT International application in the manner provided by the first paragraph of Title 35, United States Code Section 112, I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56 which became available between the filing date of the prior application and the national or PCT International filing date of application.

私は、下記の米国法典第35編120条に基づいて下記の米国特許出願に記載された権利、又は米国を指定している特許協力条約365条(c)に基づく権利をここに主張します。また、本出願の各請求範囲の内容が米国法典第35編112条第1項で規定された方法で先行する米国特許出願又は特許協力条約で規定された国際出願に開示されていない限り、その先行出願書提出日以降で本出願書の国内のまたは特許協力条約国際提出日までの期間中に入手された、連邦規則法典第37編1条56項で定義された特許資格に関する重要な情報について開示義務があることを認識しています。

(Application No.) (出願番号)	(Filing Date) (出願日)	(Status: Patented, Pending, Abandoned) (現況：特許許可済、係属中、放棄済)
(Application No.) (出願番号)	(Filing Date) (出願日)	(Status: Patented, Pending, Abandoned) (現況：特許許可済、係属中、放棄済)

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

私は、私自身の知識に基づいて本宣言書で行う表明が全て真実であり、かつ情報と信じることに基づく表明が全て真実であると信じていること、さらに故意になされた虚偽の表明及びそれと同等の行為は米国法典第18編第1001条に基づき、罰金または拘禁、もしくはその両方により処罰されること、そしてそのような故意による虚偽の声明を行えば、出願した、又は既に許可された特許の有効性が失われる可能性があることを認識し、ここに宣誓します。

POWER OF ATTORNEY: As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith: (*list name and registration number*)

委任状： 私は明記された発明者として、本出願に関する一切の手続きを米国特許商標局に対して遂行する下記の弁理士または代理人をここに指名します。(弁理士、または代理人の氏名及び登録番号を明記のこと)

Paul A. Guss (Reg. No. 33,099)

Send Correspondence to:  
(書類送付先)

Paul A. Guss  
775 South 23rd Street  
1st Floor, Suite 2  
Arlington, Virginia 22202

Direct Telephone Calls to: (name and telephone number)  
電話連絡先 (名前及び電話番号)

Paul A. Guss  
(703) 486-2710

Full name of first or sole inventor 唯一または第一発明者名 Shinya MATSUMOTO	
Inventor's signature 発明者の署名 松本 慎也	Date 日付 September 4, 2000
Residence 居所 Tokyo, JAPAN	
Citizenship 国籍 Japan	
Post Office Address 郵便宛先 c/o SONY COMPUTER ENTERTAINMENT INC., 1-1, Akasaka 7-chome, Minato-ku, Tokyo 107-0052, JAPAN	
Full name of second joint inventor, if any 第二共同発明者名 Takuya SEKI	
Second inventor's signature 第二共同発明者の署名 関 拓也	Date 日付 September 4, 2000
Residence 居所 Tokyo, JAPAN	
Citizenship 国籍 Japan	
Post Office Address 郵便宛先 c/o SONY COMPUTER ENTERTAINMENT INC., 1-1, Akasaka 7-chome, Minato-ku, Tokyo 107-0052, JAPAN	

(Supply similar information and signature for third and subsequent joint inventors.)  
(第三以降の共同発明者についても同様に記載し、署名をすること)

Full name of third joint inventor, if any 第三共同発明者名 Yuta KIUMRA	
Third inventor's signature 第三共同発明者の署名 木村 祐太	Date 日付 September 4, 2000
Residence 居所 Tokyo, JAPAN	
Citizenship 国籍 Japan	
Post Office Address 郵便宛先 c/o SONY COMPUTER ENTERTAINMENT INC., 1-1, Akasaka 7-chome, Minato-ku, Tokyo 107-0052, JAPAN	
Full name of fourth joint inventor, if any 第四共同発明者名 Yasuhiro SHIMURA	
Fourth inventor's signature 第四共同発明者の署名 志村 安啓	Date 日付 September 4, 2000
Residence 居所 Tokyo, JAPAN	
Citizenship 国籍 Japan	
Post Office Address 郵便宛先 c/o SONY COMPUTER ENTERTAINMENT INC., 1-1, Akasaka 7-chome, Minato-ku, Tokyo 107-0052, JAPAN	
Full name of fifth joint inventor, if any 第五共同発明者名	
Fifth inventor's signature 第五共同発明者の署名	Date 日付
Residence 居所	
Citizenship 国籍	
Post Office Address 郵便宛先	